

## BAB II

### LANDASAN TEORI

#### 2.1 Faktor Yang Mempengaruhi Kelulusan

Beberapa penelitian sebelumnya telah banyak mengkaji faktor-faktor yang mempengaruhi kelulusan seperti diantaranya:

1. Penelitian oleh irfan (2015), yang menganalisa pola asosiasi jalur masuk terhadap kelulusan mahasiswa dengan menggunakan metode *Fold-Growth*. Penelitian ini menyimpulkan bahwa jalur masuk yang berbeda-beda mempengaruhi tingkat kelulusan yang berbeda-beda pula yang dilihat dari IPK dan lama masa studinya.
2. Penelitian oleh Khotimah (2011) yang berjudul “Aplikasi *Data Mining* Untuk Mengukur Tingkat Kelulusan Mahasiswa Dengan Metode Apriori Dan *K-Mean Clustering*”. Penelitian ini melihat hubungan antara jalur masuk, sekolah asal, kota asal, lama masa studi dan IPK kemudian menyimpulkan bahwa terdapat hubungan antara semua atribut yang digunakan dimana jalur masuk, sekolah asal dan kota asal berpengaruh terhadap tingkat kelulusan yang dilihat dari lama masa studi dan IPK.
3. Penelitian lainnya oleh Tambun (2015) mengenai penerapan *data mining* untuk menampilkan tingkat kelulusan mahasiswa menggunakan algoritma Apriori. Atribut data yang digunakan yaitu jalur masuk, sekolah asal, kota asal, lama masa studi, dan IPK. Penelitian ini juga menyimpulkan bahwa jalur masuk, sekolah asal dan kota asal berpengaruh terhadap lama masa studi dan IPK seseorang ketika telah lulus.

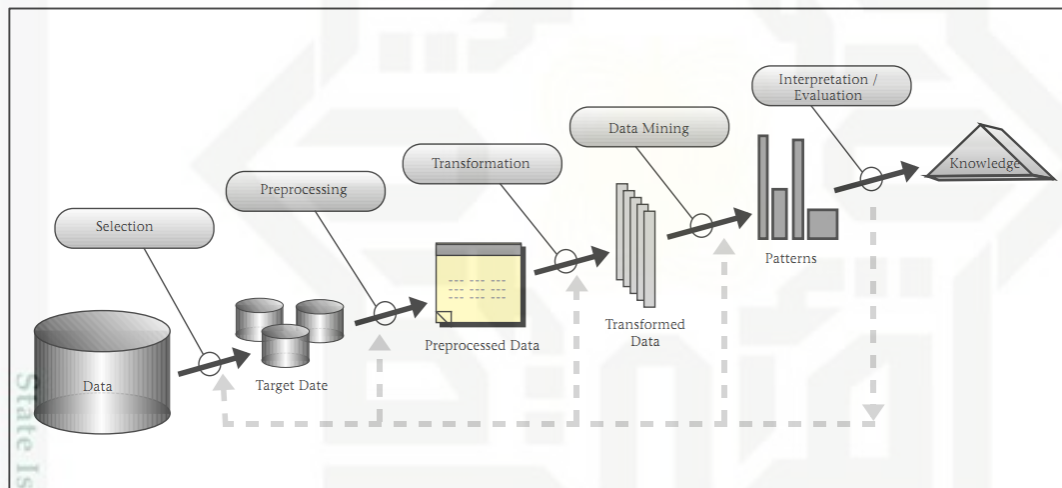
Bedasarkan beberapa penelitian di atas bisa diambil informasi bahwa data jalur masuk, sekolah asal, kota asal, berpengaruh terhadap lama masa studi serta IPK (Indeks Prestasi Kumulatif).

#### 2.2 Proses *Knowledge Discovery Database*

*Data mining* juga dikenal dengan istilah *Knowledge Discovery in Database* (KDD). Menurut Gorunescu (2011) yang dikutip dari Prasetyo (2014) *data mining* terdiri dari 3 langkah utama yaitu:

1. Eksplorasi atau pemrosesan data awal. Tahap ini terdiri dari pembersihan data, normalisasi data, transformasi data dan sebagainya.
2. Membangun model dan melakukan validasi terhadapnya. Tahap ini ialah pemilihan fungsi dari *data mining* seperti klasifikasi, *cluster*, analisis pola asosiasi dan sebagainya.
3. Penerapan. Tahap ini penerapan *knowledge* atau informasi pengetahuan yang didapatkan dari tahap sebelumnya.

Sedangkan menurut Fayyad (1996), proses KDD terdiri dari serangkaian proses yang saling terhubung dan diselesaikan secara berurutan. Proses-proses tersebut terdiri *selection*, *preprocessing*, *transformation*, *data mining* dan terakhir *evaluation*. Kemudian menurut Hermawati (2013) rangkaian dari tahapan proses KDD diilustrasikan dengan Gambar 2.1 berikut ini:



Gambar 2.1 Proses KDD

### 2.2.1 Selection

Proses seleksi adalah proses memilih data apa saja yang akan digunakan. Data diseleksi dari satu atau beberapa *database* kemudian dimasukkan ke sebuah *database* baru. Proses seleksi dilakukan karena terkadang tidak semua data yang ada di dalam *database* dibutuhkan untuk proses *data mining*.

### 2.2.2 Preprocessing

Data yang sudah di seleksi kemudian dilakukan pembersihan data (*cleanning data*) dan integrasi data (*integration data*). *Cleanning* dilakukan untuk menghapus data yang memiliki *field* kosong, memiliki kesalahan cetak atau

kesalahan penulisan (*typography*) dan mencegah data yang redundan atau data kembar. Sedangkan *integration* adalah proses penggabungan atribut jika sebelumnya terpisah-pisah kemudian dimasukan ke sebuah *database* baru.

### 2.2.3 Transformation

*Transformation* adalah proses merubah atau menggabungkan data ke dalam format yang sesuai untuk di proses dalam *data mining*. Transformasi data dapat dilakukan dengan cara-cara berikut :

- Smoothing*, bekerja untuk menghilangkan *noise* dari data.
- Attribute construction*, atribut baru dibuat atau ditambahkan untuk membantu proses *mining*.
- Agregation*, ringkasan atau operasi agrerasi diterapkan pada data.
- Normalization*, data atribut dibuat dalam skala tertentu sehingga menjadi kisaran data yang lebih kecil sehingga sebaran datanya tidak terlalu jauh. Beberapa metode yang bisa dipakai pada proses normalisasi seperti berikut ini:

- 1) *Min-Max*, metode ini melakukan transformasi linear terhadap data. Dimana rumusnya sebagai berikut:

$$Data\ Baru = \frac{(Data - Minimum) * (Maksimum\ Baru - Minimum\ Baru)}{(Maksimum - Minimum) + Minimum\ Baru} \quad (2.1)$$

- 2) *Z-Score*, metode ini ialah normalisasi yang berdasarkan *mean* (nilai rata-rata) dan *standard deviation* (deviasi standar) dari data. Dimana rumusnya sebagai berikut:

$$Data\ Baru = \frac{Data - Mean}{Nilai\ Standar\ Divisiasi} \quad (2.2)$$

- 3) *Discretization*, dimana data yang semula dalam bentuk nilai angka diubah menjadi data dengan interval label. Seperti data IPK di ubah menjadi kategori rendah, sedang dan tinggi
- 4) *Logika Fuzzy*, metode ini digunakan untuk membantu menyelesaikan permasalahan *discretization* data dimana nilai dari data terdapat pada nilai keabuan atau memiliki dua kategori. Misalnya IPK 2,7 masuk ke kategori rendah dan sedang. Fungsi dari logika *fuzzy* adalah mencari pendekatan nilai tersebut masuk ke kategori yang mana menggunakan persamaan-persamaan yang dijelaskan pada sub bab logika *fuzzy*.

### 2.2.3.1 Logika Fuzzy

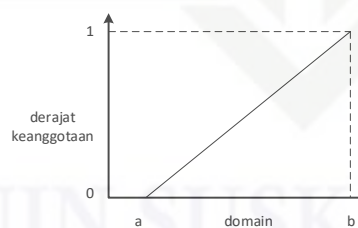
Pada tahap transformasi ini, digunakan teknik lain untuk mengolah data sebelum dilanjutkan kepada proses *data mining*. Teknik yang digunakan adalah suatu model persamaan matematika untuk mengolah data pada saat proses normalisasi. Model persamaan matematika yang akan digunakan bernama logika *fuzzy*.

Menurut Sri Kusumadewi (2013), Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika *fuzzy*.

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya atau disebut juga derajat keanggotaan yang memiliki interval antara 0 hingga 1. Salah satu cara yang dapat digunakan untuk mendapatkan derajat keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan diantaranya:

#### a. Representasi Linear

Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama kenaikan himpunan dimulai dari domain yang memiliki derajat keanggotaan bernilai 0 hingga bergerak naik ke derajat keanggotaan yang bernilai 1.



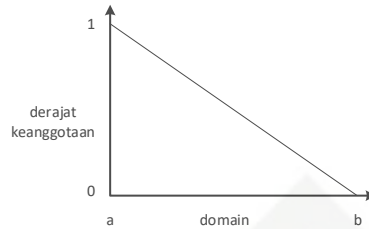
**Gambar 2.2 Representasi Linear Naik**

fungsi keanggotaannya adalah sebagai berikut:

$$\mu[x] = \begin{cases} 0 & \text{jika } x \leq a \\ \frac{x-a}{b-a} & \text{jika } a \leq x \leq b \\ 1 & \text{jika } x \geq b \end{cases} \quad (2.3)$$



Kemudian yang kedua merupakan kebalikan dari yang pertama. Dimulai dari domain yang memiliki derajat keanggotaan 1, bergerak menurun ke derajat keanggotaan bernilai 0.



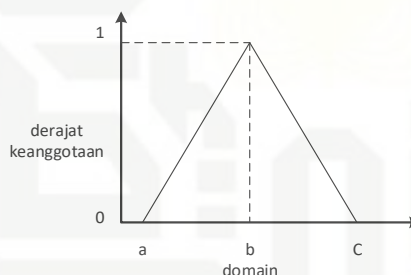
**Gambar 2.3 Representasi Linear Turun**

fungsi keanggotaannya adalah sebagai berikut:

$$\mu[x] = \begin{cases} \frac{b-x}{b-a} & \text{jika } a \leq x \leq b \\ 0 & \text{jika } x \geq b \end{cases} \quad (2.4)$$

b. Representasi Kurva Segitiga

Kurva segitiga sebenarnya adalah gabungan dari dua garis (linear) di atas. Bentuk dari kurva segitiga dapat dilihat dari Gambar 2.4 berikut ini:



**Gambar 2.4 Representasi Kurva Segitiga**

fungsi keanggotaannya seperti berikut:

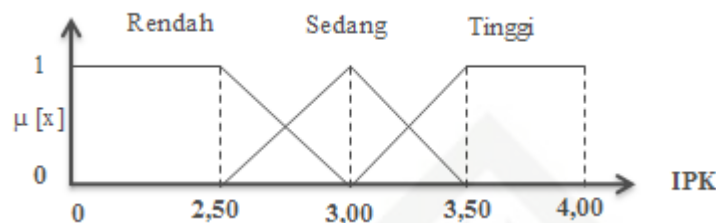
$$\mu[x] = \begin{cases} 0 & \text{jika } x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a} & \text{jika } a \leq x \leq b \\ \frac{b-x}{c-b} & \text{jika } b \leq x \leq c \end{cases} \quad (2.5)$$

Dari penjelasan di atas maka dilakukan normalisasi terhadap atribut IPK menggunakan logika *fuzzy*. Dimana atribut IPK diubah menjadi 3 kategori sebagai berikut:

1. Kategori rendah yaitu IPK dari 0 sampai 3
2. Kategori sedang yaitu IPK dari 2,5 sampai 3,5

### 3. Kategori tinggi yaitu IPK dari 3 sampai 4

Semua kategori tersebut dihitung menggunakan fungsi keanggotaan logika fuzzy. Fungsi keanggotaan dari setiap himpunan dapat dilihat dari Gambar 2.5 berikut ini:



**Gambar 2.5 Kurva Fuzzy Kategori IPK**

Kemudian fungsi keanggotaan untuk setiap himpunannya dapat dilihat pada persamaan-persamaan berikut ini:

$$\mu_{\text{Rendah}}[X] = \begin{cases} 1 & \text{jika } x \leq 2,5 \\ \frac{3-x}{3-2,5} & \text{jika } 2,5 \leq x \leq 3 \\ 0 & \text{jika } x \geq 3 \end{cases} \quad (2.6)$$

$$\mu_{\text{Sedang}}[X] = \begin{cases} 0 & \text{jika } x \leq 2,5 \text{ atau } x \geq 3,5 \\ \frac{x-2,5}{3-2,5} & \text{jika } 2,5 \leq x \leq 3 \\ \frac{3,5-x}{3,5-3} & \text{jika } 3 \leq x \leq 3,5 \end{cases} \quad (2.7)$$

$$\mu_{\text{Tinggi}}[X] = \begin{cases} 0 & \text{jika } x \leq 3 \\ \frac{x-3}{3,5-3} & \text{jika } 3 \leq x \leq 3,5 \\ 1 & \text{jika } x \geq 3,5 \end{cases} \quad (2.8)$$

#### 2.2.4 Data Mining

*Data mining* adalah istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. *Data mining* adalah proses yang menggunakan teknik statistika, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terakut dari berbagai *database* besar (E. Turban, 2005).

Pengertian lain menurut Larose (2005), *data mining* adalah bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistika, *database* dan visualisasi untuk penanganan permasalahan pengambilan informasi dari *database* yang besar. Sedangkan menurut Beynon (2004) *data mining* adalah penambangan atau penemuan

informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar.

Dari beberapa pendapat para ahli tersebut maka dapat ditarik suatu kesimpulan bahwa *data mining* adalah suatu cabang ilmu pengetahuan yang menggunakan teknik-teknik tertentu yang bertujuan untuk mendapatkan informasi pengetahuan dari sekumpulan data yang besar.

#### 2.2.4.1 Karakteristik *Data Mining*

Menurut Beynon (2004), *data mining* memiliki beberapa karakteristik seperti berikut ini:

1. *Data mining* berhubungan dengan proses penemuan sesuatu atau pola data tertentu yang tersembunyi atau tidak diketahui sebelumnya.
2. *Data mining* bisa digunakan untuk mengolah data dalam jumlah yang sangat besar
3. *Data mining* biasanya dibutuhkan untuk membuat keputusan penting atau keputusan strategis

#### 2.2.4.2 Pengelompokan *Data Mining*

*Data mining* dapat dikelompokkan menjadi beberapa kelompok berdasarkan tugas dan fungsinya masing-masing (Larose, 2005), berikut penjelasannya:

##### 1. Deskripsi

*Data mining* dapat digunakan untuk mencari deskripsi dari pola atau kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara tentu tidak dapat langsung mengetahui keterangan siapa calon yang akan mendapatkan dukungan paling sedikit atau paling banyak pada pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

##### 2. Klasifikasi

Didalam klasifikasi terdapat target variabel kategori. Fungsi dari klasifikasi adalah untuk mengklasifikasikan suatu target *class* ke dalam kategori yang dipilih tersebut. Salah satu algoritma data mining untuk fungsi klasifikasi adalah algoritma C4.5, contoh penelitian yang menerapkan algoritma ini berjudul “Aplikasi *Mining Data* Mahasiswa Dengan Metode Klasifikasi *Decision Tree*”

oleh (Sunjana, 2010) yang menghasilkan klasifikasi dari matakuliah yang mempunyai pengaruh signifikan terhadap IPK mahasiswa. Contoh algoritma lainnya yang dipakai menyelesaikan masalah klasifikasi yaitu algoritma KNN dan M-KNN.

### 3. Klustering

Pengklasteran merupakan pengelompokan *record*, pengamatan atau memperhatikan dan membentuk *class-class* objek yang membentuk kemiripan. *Cluster* adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan tidak memiliki kemiripan dengan *record-record* dengan *cluster* yang lain.

### 4. Prediksi

Prediksi hampir sama dengan klasifikasi, kecuali dalam prediksi nilai dari hasil akan ada pada masa yang akan datang. Contoh kasus prediksi yaitu prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan dalam prediksi tetapi tergantung dari kasusnya.

### 5. Asosiasi

Menurut Fahrian (2008), asosiasi adalah aturan yang digunakan untuk melihat hubungan di antara data atau bagaimana suatu kelompok data mempengaruhi suatu keberadaan data. Pengertian lainnya asosiasi merupakan suatu aturan atau prosedur untuk mencari nilai hubungan satu *item* dengan *item* lainnya dalam satu data set.

Asosiasi dalam *data mining* adalah menemukan keterkaitan atribut yang muncul dalam satu waktu. Asosiasi dalam *data mining* lebih umum dikenal dengan istilah *market basket analysis*, karena fungsinya adalah untuk mencari keterkaitan antara atribut atau yang biasa dikenal dengan *itemset*, berdasarkan jumlah *itemset* yang muncul dan direpresentasikan ke dalam bentuk *association rules*. Agar lebih mudah memahami istilah-istilah dalam asosiasi, berikut ini ada sebuah contoh tabel berisi sekumpulan data yang akan di analisa menggunakan teknik asosiasi:



**Tabel 2.1 Contoh Dataset**

No.Transaksi	Item
1	A , D , G, H
2	B , D
3	A , D
4	D , G
5	D , G , H

Semua data yang dianalisa menggunakan teknik asosiasi adalah data dari masa lampau atau biasa disebut dengan data historis. Teknik ini digunakan untuk mempelajari, menemukan atau menggali informasi yang berharga dari data histori tersebut. Informasi yang didapatkan di representasikan berupa aturan-aturan asosiasi. Dari tabel 2.1 bisa dilihat terdapat beberapa *item* yaitu A, B, D, G, dan H. Himpunan dari beberapa *item* dilambangkan dengan I. Seperti Persamaan 2.9 berikut:

$$I = \{ A, B, D, G, H \} \quad (2.9)$$

Dalam asosiasi ada dua tahap yang dilakukan, yaitu pencarian *frequent itemset* (jumlah transaksi yang mengandung *item* x). Contohnya dari tabel 2.1 pencarian *frequent item* (A). *Item* (A) memiliki *frequent item*=2, karena *item* (A) terdapat pada transaksi 1 dan 3 (A)={1,3}. Setelah semua *frequent* satu (1) *item* dicari, selanjutnya dilakukan kombinasi 2 *item* dan iterasi hingga *k-itemset*. Contohnya pencarian *frequent itemset* (A,D)=2 dan *frequent itemset* (D,G,H)=2. Gabungan dari beberapa *item* disebut dengan *itemset*. Kemudian persentase dari transaksi yang mengandung *item* x disebut dengan nilai *support* atau nilai pendukung *item* x. Setelah menemukan berapa jumlah transaksi yang mengandung *item* x maka barulah yang kedua dicari *association rules*-nya.

Jika nilai *support* suatu *itemset* lebih besar atau sama dengan minimum *support*  $\sigma$  (*threshold*), maka *itemset* tersebut dikatakan *frequent itemset* dan yang tidak memenuhi dinamakan *infrequent*. Nilai dari minimum *support*  $\sigma$  (*threshold*) ditentukan oleh pengguna. Nilai tersebut adalah batas minimum jumlah *item* diperbolehkan untuk dibentuk ke dalam *association rules*. Untuk mendapatkan nilai *support* maka dapat dicari menggunakan Persamaan 2.10 berikut:

$$Support(x) = \frac{Frequent\ Itemset\ X}{Total\ Transaksi} \quad (2.10)$$

Sedangkan untuk mencari nilai *support* dari dua *item* dapat menggunakan Persamaan 2.11 berikut:

$$Support(x \cap y) = \frac{Frequent\ Itemset\ X \cap Y}{Total\ Transaksi} \quad (2.11)$$

Setelah semua *frequent item* didapat, maka ditentukan syarat minimum *confidence*. *Confidence* adalah nilai ukuran seberapa besar valid tidaknya suatu *association rules*. *Confidence* adalah proporsi dari semua transaksi yang mengandung baik (x) maupun (y) dengan yang mengandung (x). Persamaan untuk mencari nilai *confidence* berdasarkan Persamaan 2.12 berikut:

$$Confidence(x \rightarrow y) = \frac{Frequent\ Itemset\ X \cap Y}{Frequent\ Itemset\ X} \quad (2.12)$$

#### 2.2.4.3 Algoritma *Equivalence Class Transformation*

Algoritma ECLAT (*Equivalence Class Transformation*) merupakan salah satu algoritma yang digunakan untuk memecahkan kasus-kasus asosiasi dalam data *mining*. Algoritma ini sangat sederhana untuk menemukan *itemset* yang paling sering muncul atau disebut juga *frequent itemset*, pada dasarnya algoritma ECLAT melakukan pencarian secara *depth-first search* pada *database* dengan tata letak vertikal, jika *database* berbentuk horizontal maka harus dikonversikan ke bentuk vertikal terlebih dahulu (Kaur, 2014).

Kemudian menurut Aakash & Gadhiya (2014), proses pembentukan *itemset* pada algoritma ECLAT di mulai dengan mengubah bentuk transaksi, jika *item* pada transaksi berbentuk horizontal maka diubah menjadi bentuk vertikal dengan menggabungkan *TID list* pada transaksi yang memiliki *item* yang sama.

Dapat disimpulkan algoritma ECLAT merupakan salah satu algoritma asosiasi yang digunakan untuk mencari *frequent itemset* (jumlah setiap *item* yang terdapat didalam transaksi *database*). Proses pencarian *item* oleh algoritma ini sederhana dengan melakukan sekali *scanning database* secara *depth first search*. Kemudian hasilnya direpresentasikan ke dalam format vertikal yang bertujuan memudahkan untuk mencari kombinasi dari *item* berikutnya. *Item* berserta simpulnya disimpan bersama di sebuah *TID list* (*Transaction ID list*). Proses *scanning database* yang membedakan algoritma ECLAT dengan algoritma Apriori dan *Fold Growth*.

Berikut ini merupakan langkah-langkah mencari *frequent itemset* menggunakan algoritma ECLAT :

1. Lakukan *scanning database* untuk menemukan semua kandidat *item*
2. Representasikan data ke dalam bentuk format data vertikal.
3. Tentukan nilai minimum *support*.
4. Cari data kandidat *k-itemset* (Ck) lalu hitung masing-masing nilai *support*-nya.
5. Hilangkan *itemset* yang nilai *support*-nya kurang dari nilai minimum *support*.
6. Setelah mendapatkan data yang memenuhi nilai minimum *support*, lakukan kombinasi pada *itemset* berdasarkan TID-set yang sama pada data-data tersebut sehingga menciptakan *k-itemset*.
7. Cari *association rules* dengan cara menentukan nilai *confidence*-nya.
8. Hilangkan data yang nilai *confidence*-nya kurang dari nilai minimum *confidence*.

Selanjutnya *pseudocode* dari algoritma ECLAT dapat dilihat pada Gambar

2.6 berikut ini (Schmidt-Thieme, 2003):

```

Algoritma Eclat
//Initial Call:  $F \leftarrow \emptyset$  ;  $P \leftarrow \{(i, t(i)) | i \in I, |t(i)| \geq \text{minsup}\}$ 
ECLAT (P,minsup,F)
Foreach  $[X_a, t(X_a)] \in P$  do
     $F \leftarrow F \cup \{ (X_a, \text{sup}(X_a) \in P) \}$ 
     $P \leftarrow \emptyset$ 
    Foreach  $[X_b, t(X_b)] \in P$  ; with  $X_b > X_a$  do
         $X_{ab} = X_a \cup X_b$ 
         $t(X_{ab}) = t(X_a) \cap t(X_b)$ 
        If  $\text{sup}(X_{ab}) \geq \text{minsup}$  then
             $P_a \leftarrow P_a \cup \{ [X_{ab}, t(X_{ab})] \}$ 
        Endif
    Endfor
    If  $P_a \neq \emptyset$  then ECLAT( $P_a, \text{minsup}, F$ )
Endfor
    
```

keterangan:

P : *Prefix Equivalence Class*  
Minsup : *Minimum Support*  
F : *Frequent Itemset*

**Gambar 2.6 Pseudocode Algoritma ECLAT**



### 2.2.5 Evaluation

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian proses yang juga disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis sebelumnya.

## 2.3 Kajian Pustaka

Beberapa penelitian terkait adalah penelitian oleh Irfan (2015) yang berjudul “Analisa Pola Asosiasi Jalur Masuk Terhadap Kelulusan Mahasiswa dengan Metode *Flod Growth*”. Dari penelitian ini didapat hasil mahasiswa yang mengikuti jalur masuk tes tulis pada Jurusan Teknik Informatika dengan lama lulusnya lebih dari 4 tahun nilai support-nya 28,73. Selanjutnya penelitian oleh Tambun (2015) yang berjudul “Rancang Bangun Aplikasi *Data Mining* Untuk Menampilkan Informasi Kategori IPK Mahasiswa Dengan Algoritma Apriori”. Kesimpulannya algoritma apriori dapat diterapkan dan menampilkan informasi tingkat kelulusan mahasiswa berupa nilai *support* dan nilai *confidence*.

Selain itu beberapa penelitian lain yang menunjang penelitian ini yang berjudul “*Identification Of Best Algorithm In Association Rule Mining Based On Performance*” penelitian ini membandingkan performa dari tiga algoritma asosiasi yaitu FP-growth, apriori dan ECLAT. Hasil pada penelitian ini menunjukkan algoritma ECLAT merupakan algoritma terbaik dari ketiganya, karena memiliki waktu eksekusi yang lebih cepat dengan nilai *support* dan *confidence* yang lebih baik (Sinha, 2014).

Penelitian lain oleh Kaur (2014) yang berjudul “*ECLAT Algorithm for Frequent Itemsets Generation*” menyimpulkan hasil yang sama dengan penelitian oleh Sinha (2014) bahwa algoritma ECLAT lebih baik daripada algoritma apriori dalam menekan waktu eksekusi.